

# Esame di Calcolatori Elettronici T

## 21 Dicembre 2018 (Ing. Informatica)

### Esercizio 1

Si progetti un *watchdog timer* per sistema basato sul processore DLX a **1 GHz** dotato di **768 MB di EPROM** mappata negli indirizzi bassi e **2 GB di RAM** mappata negli indirizzi alti. Il dispositivo *watchdog*, inizialmente non abilitato, una volta in funzione deve **resettare l'intero sistema se non è riavviato entro un tempo T** programmabile via software, ma che **non può superare i 4 sec.** Il DLX deve essere avvisato dell'imminente reset del sistema **quando rimane un tempo pari a T/2** e attivare di conseguenza la procedura di riavvio del *watchdog* mediante opportune istruzioni. Fintantoché il *watchdog* è abilitato, la procedura appena descritta deve essere ripetuta continuamente. Si assuma che la "**Rete di RESET**" (già progettata e mostrata in figura), che asserisce il segnale di **RESET** dell'intero sistema, attivi tale procedura se riceve sul proprio ingresso **TIME\_OUT** un valore logico 1 per almeno **3 clock consecutivi**.



- Per prima cosa, descrivere sinteticamente la soluzione che s'intende realizzare e indicare chiaramente quali sono i segnali di *chip-select*
- Progettare il sistema, **minimizzando le risorse necessarie**
- Indicare le espressioni di decodifica e il range di indirizzi di tutte le periferiche, le memorie e i segnali
- Scrivere il codice che consente di abilitare e programmare il *watchdog* con  $T=1$  sec e il codice dell'interrupt handler che consente il riavvio del *watchdog* con la medesima programmazione (i.e.,  $T = 1$  sec)
- Si faccia l'ipotesi che i registri da R25 a R29 possano essere utilizzati senza la necessità di doverli ripristinare durante l'esecuzione degli interrupt handler

### Esercizio 2

Con riferimento all'ISA del DLX studiata nel corso:

- a) E' possibile annidare le interruzione? **Si o No?**
- b) Se sì, **come?** Se no, **perché?**

### Esercizio 3

Quali sono e come possono essere eliminate le alee strutturali nel DLX pipelined?



`RAM` :  $0 \times 8000\ 0000 \rightarrow 0 \times FFFF\ FFFF$  (4 x 512 MB)  
`EPROM_H` :  $0 \times 2000\ 0000 \rightarrow 0 \times 2FFF\ FFFF$  (4 x 64)  
`EPROM_L` :  $0 \times 0000\ 0000 \rightarrow 0 \times 1FFF\ FFFF$  (4 x 128)

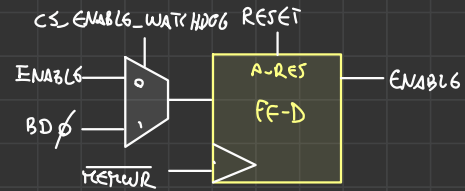
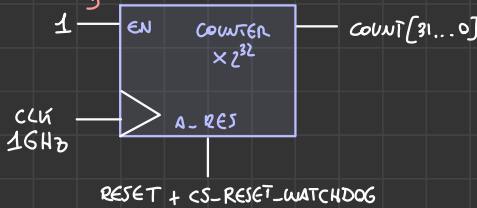
`CS_RAM_0` = BA31 BA60  
`CS_RAM_1` = BA31 BE1  
`CS_RAM_2` = BA31 BE2  
`CS_RAM_3` = BA31 BE3

`CS_ENABLE_WATCHDOG` = BA31 BA30 BA2 BE0  
`CS_RES_WATCHDOG` = BA31 BA30 BA2 BE1 · `MEMWR`  
`CS_SET_RANGE` = BA31 BA30 BA2

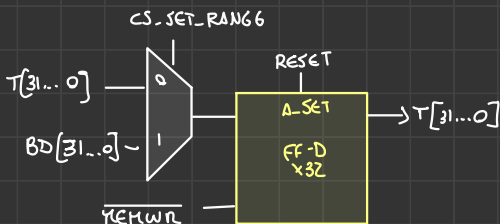
`CS_EPROM_H_0` = BA31 BA30 BA29 BE0  
`CS_EPROM_H_1` = BA31 BA30 BA29 BE1  
`CS_EPROM_H_2` = BA31 BA30 BA29 BE2  
`CS_EPROM_H_3` = BA31 BA30 BA29 BE3

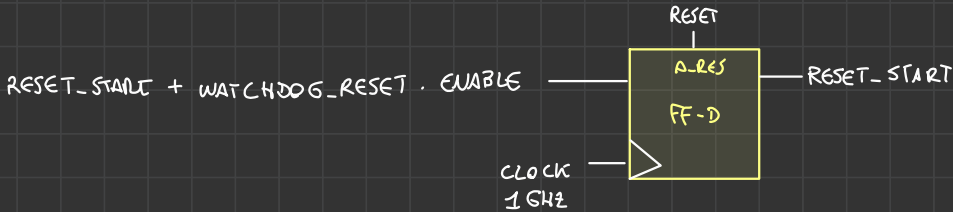
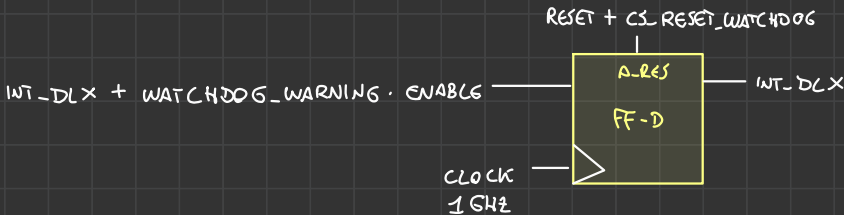
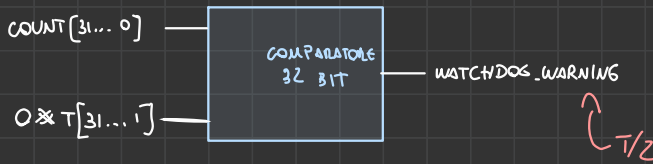
`CS_EPROM_L_0` = BA31 BA30 BA29 BE0  
`CS_EPROM_L_1` = BA31 BA30 BA29 BE1  
`CS_EPROM_L_2` = BA31 BA30 BA29 BE2  
`CS_EPROM_L_3` = BA31 BA30 BA29 BE3

Per il conteggio di massimo 4 sec.



Corrente di abilitare o disabilitare il watchdog





```

100 0h  LHI    R25, 0x4000 ; R25 = 0x4000 0000
100 4h  SH     R0, 0x0000(R25); disabilita watchdog e resetta counter
100 8h  LHI    R26, 0x3B9A ; T = (10^9 - 1) / 10 = 0x3B9AC9FF
100 Ch  ADDU   R26, R26, 0xC9FF; imposta conteggio a CS_SET_RANGE
100 10h SW     R26, 0x0004(R25); R27 = 0x0000 0001
100 14h ADDI   R27, 0x0001(R0); abilita watchdog e resetta counter
100 18h SH     R27, 0x0000(R25); ritorna all'header (8h)
100 1Ch JR     R31;

```

```

0h  HANDLER: ADDI R28, R0, 0x1000
4h  JAL R28 (esegue il codice di sopra)
8h  RFE

```